MARYLAND

TECHNICAL RESEARCH REPORT

SRC TR 85-13-r1

A Superlinearly Convergent Feasible Method for the Solution of Inequality Constrained Optimization Problems

by

E.R. Panier and A.L. Tits

SYSTEMS RESEARCH CENTER

UNIVERSITY OF MARYLAND

COLLEGE PARK, MARYLAND 20742

SIAM J. CONTROL AND OPTIMIZATION
Vol. 25, No. 4, July 1987

006

A SUPERLINEARLY CONVERGENT FEASIBLE METHOD FOR THE SOLUTION OF INEQUALITY CONSTRAINED OPTIMIZATION PROBLEMS*

ELIANE R. PANIER† AND ANDRÉ L. TITS†

Abstract. When iteratively solving optimization problems arising from engineering design applications, it is sometimes crucial that all iterates satisfy a given set of "hard" inequality constraints, and generally desirable that the objective function value improve at each iteration. In this paper, we propose an algorithm of the successive quadratic programming (SQP) type which, unlike other algorithms of this type, does enjoy such properties. Under mild assumptions, the new algorithm is shown to converge from any initial point, locally superlinearly. Numerically tested, it has proven to be competitive with the most successful currently available nonlinear programming algorithms, while the latter do not exhibit the desired properties.

Key words. constrained optimization, successive quadratic programming, superlinear convergence, engineering design

AMS(MOS) subject classifications. 90C30, 65K10

1. Introduction. While some of the specifications associated with engineering design problems can often be relaxed, others, such as stability or physical realizability, have to be met imperatively (see [13] for a discussion of optimization problems arising from design problems). The former type of specification calls for tradeoff exploration through close interaction between designer and design process. However, this tradeoff exploration can meaningfully take place only once the latter specifications are satisfied. Since each iteration of an optimization algorithm involves one or more function evaluations and since typically, in a design environment, function evaluations call for computationally expensive system simulations, it is essentially required that hard constraints be satisfied *at each iteration.*[1] It is also desirable that the design obtained after each iteration improve on the previous one.

In the simplest case, a design problem can be formulated as

$$(\mathrm{P})\begin{cases}\min f(x)\\ \text{s.t. } x\in X\end{cases}$$

where $X=\{x \text{ s.t. } g_j(x)\leqq 0, j=1,\cdots m\}$ and $f:\mathbb{R}^n\to\mathbb{R}$ and $g_j:\mathbb{R}^n\to\mathbb{R}$, $j=1,\cdots m$, are smooth functions. For this optimization problem, the stipulations outlined above amount to the requirement that, given $x_0\in X$, the optimization algorithm construct a sequence $\{x_k\}_{k=0}^{\infty}$ such that, for all k,

$$x_k\in X \tag{1.1}$$

and

$$f(x_{k+1})\leqq f(x_k). \tag{1.2}$$

* Received by the editors August 9, 1985; accepted for publication (in revised form) May 13, 1986. This work was supported by National Science Foundation grants DMC-84-20740 and CDR-85-00108, by a grant from the Minta Martin Foundation, College of Engineering, University of Maryland, by a grant from the Engineering Research Center at the University of Maryland, and by a grant from the Westinghouse Corporation.

† Electrical Engineering Department and Systems Research Center, University of Maryland, College Park, Maryland 20742.

[1] In fact some simulation programs (such as SPICE2 [10]) will refuse parameter values which violate some physical realizability constraints.

Methods of feasible directions [23], [14] satisfy these two requirements. They have been extended to handle problems with functional constraints [3], [15] and multiple objectives [13] and enhanced to efficiently handle design problems [21]. They have been used very successfully in solving engineering design problems arising in diverse application areas [12], [1], [11]. However, they suffer from an important shortcoming in that they are generally *slow*, as their rate of convergence is at best linear.

This paper presents an algorithm which enjoys properties (1.1) and (1.2) as well as a superlinear rate of convergence. This algorithm is of the successive quadratic programming (SQP) type. Successive quadratic programming algorithms were first introduced by Wilson [22]. Subsequently, Robinson [20] showed that Wilson's method is locally quadratically convergent and that it can be viewed as a form of Newton's method for solving the first order necessary conditions of optimality for constrained nonlinear programming problems. The question of global convergence and Hessian approximation were then considered by a number of authors (see e.g. [4], [2]). Numerical experiments have shown that these methods (in particular a version due to Powell [17]) often dramatically outperform algorithms of other classes [6]. However, existing SQP type algorithms do not enjoy properties (1.1) and (1.2).

Given an estimate $x \in X$ of the solution x^* to problem (P) and an estimate H of the Hessian of the Lagrangian at x^*, the SQP iteration yields a search direction d^0 given by the solution of the quadratic program

$$\begin{aligned}&\min \tfrac{1}{2} d^T H d + \langle \nabla f(x), d\rangle \\ &\text{s.t. } g_j(x) + \langle \nabla g_j(x), d\rangle \leqq 0, \qquad j = 1, \cdots m.\end{aligned} \tag{1.3}$$

Let us assume for the time being that H is positive definite. Then clearly d^0 is a descent direction for f at x, since, using the first order condition of optimality for (1.3), we get

$$\begin{aligned}\langle \nabla f(x), d^0\rangle &= -\langle Hd^0, d^0\rangle - \sum_j \mu_j \langle \nabla g_j(x), d^0\rangle \\ &= -\langle Hd^0, d^0\rangle + \sum_j \mu_j g_j(x) \\ &\leqq -\rho |d^0|^2\end{aligned} \tag{1.4}$$

for some positive ρ and some nonnegative multipliers μ_j. However d^0 may not be a feasible direction at x, since the constraints in (1.3) merely imply, for the constraints active at x,

$$\langle \nabla g_j(x), d^0\rangle \leqq 0$$

and thus property (1.1) may not be satisfied. Feasibility is recovered if one substitutes in the right-hand side of the constraints of (1.3) a negative number $-\varepsilon$. However the new solution d^1 may not be any more a descent direction for f, thus jeopardizing property (1.2). Indeed, (1.4) now becomes

$$\langle \nabla f(x), d^1\rangle \leqq -\rho |d^1|^2 + \varepsilon \sum_j \mu_j.$$

Choosing $\varepsilon = |d^1|^\nu$, with $\nu > 2$, would resolve this difficulty, at least for d^1 small, which is the case if x is close to the solution of (P). Unfortunately the transformed problem would not be a quadratic program any more. Following an idea used by Herskovits in a different context [5], we propose to solve successively two quadratic programs: first (1.3), giving d^0, then

$$\begin{aligned}&\min \tfrac{1}{2} d^T H d + \langle \nabla f(x), d\rangle \\ &\text{s.t. } g_j(x) + \langle \nabla g_j(x), d\rangle \leqq -|d^0|^\nu, \qquad j = 1, \cdots m\end{aligned}$$

yielding a search direction d^1. The hope is that d^1 will be small enough, and close enough to d^0, near the solution to (P), for property (1.2) as well as the basic convergence properties of the SQP type algorithms to be preserved. As shown in a later section, such will indeed be the case, even without assuming positive definiteness of H over the entire space (the milder assumption (4.1) will be used instead).

Once a feasible descent direction is obtained, an Armijo type rule may be suitable as a line search procedure. However, in order to preserve a superlinear rate of convergence, it is necessary to avoid any Maratos-like effect [7], by which the step length is truncated even close to the solution. Mayne and Polak [9] solve this problem in a different context—SQP methods using a penalty function for the stepsize calculation—by replacing the line search by a search along a suitably defined arc, tangent to d^1 at x. In our context, a further "bending" towards the feasible region is necessary to avoid truncation of the step due to infeasibility. It turns out that the amount of bending must be closely monitored. Indeed, the bent unit step, say, $d^1+\tilde{d}$, must be very close to d^1 when d^1 is small (in the neighborhood of a solution of (P)). Otherwise, $d^1+\tilde{d}$ may not inherit enough descent properties from d^1, resulting again in a truncated step. Also, if $\tilde{d}$ is too large, even the unit step iteration may not yield superlinear convergence. A suitable correction $\tilde{d}$ will be obtained as the solution of a linear least squares problem.

The last problem to be addressed is that of global convergence. As suggested above, d^1 is guaranteed to be a descent direction for f only in the neighborhood of a solution to (P). Away from a solution, a first order search direction will be used. A suitable mechanism will ensure that our algorithm selects the SQP direction when a solution is approached, so that superlinear convergence can occur.

The resulting algorithm is relatively complex, as it involves the solution of two quadratic programs and of one linear least squares problem at most iterations. Clearly however, the close relationship between these three problems should result, in a clever implementation, in little more computational effort than that required for the solution of a single quadratic program.

The remainder of this paper is organized as follows. The proposed algorithm is stated in § 2. In § 3, it is shown that, under mild assumptions, this algorithm is convergent irrespective of the initial guess. Rate of convergence analysis is the object of § 4, where conditions for superlinear convergence are put forth. Finally, § 5 is devoted to implementation aspects and to numerical experiments.

2. The algorithm. Throughout the paper, the following two hypotheses will be assumed to hold.

H1. The set X is not empty;

H2. The functions f, g_j, $j=1,\cdots m$ are continuously differentiable.

The algorithm we propose for solving (P) is as follows.

ALGORITHM A.

Parameters.

$$M>0, \quad \alpha\in(0,\tfrac{1}{2}), \quad \beta\in(0,1), \quad \nu>2, \quad \kappa>2, \quad \tau\in(2,3).$$

Data.

$$x_0\in X, \qquad H_0\in\mathbb{R}^{n\times n}.$$

Step 0. *Initialization.*

Set $k=0$.

Step. 1. *Computation of a search direction.*

(i) Solve

$$(\mathrm{QP}_0)\begin{cases}\min \frac{1}{2}d^T H_k d + \langle \nabla f(x_k), d\rangle \\ \text{s.t. } g_j(x_k) + \langle \nabla g_j(x_k), d\rangle \leqq 0, \qquad j=1,\cdots m\end{cases}$$

to the extent of obtaining a Kuhn-Tucker point d_k^0 of least norm.
If (QP_0) has no Kuhn-Tucker point or if $|d_k^0| > M$ or if $|H_k d_k^0| > |d_k^0|^{1/2}$, go to (iv).

If $|d_k^0| = 0$ stop.

(ii) Solve

$$(\mathrm{QP})\begin{cases}\min \frac{1}{2}d^T H_k d + \langle \nabla f(x_k), d\rangle \\ \text{s.t. } g_j(x_k) + \langle \nabla g_j(x_k), d\rangle \leqq -|d_k^0|^\nu, \qquad j=1,\cdots m\end{cases}$$

to the extent of obtaining a Kuhn-Tucker point d_k of least norm.
If d_k exists, set $\theta_k = \langle \nabla f(x_k), d_k\rangle$.
If (QP) has no Kuhn-Tucker point or if $|d_k| > M$ or if $\theta_k > \min(-|d_k^0|^\kappa, -|d_k|^\kappa)$, go to (iv).

(iii) Compute a correction $\tilde{d}_k$, solution of the linear least squares problem

$$(\mathrm{LS})\begin{cases}\min \frac{1}{2}|d|^2 \\ \text{s.t. } g_j(x_k + d_k) + \langle \nabla g_j(x_k), d\rangle = -|d_k^0|^\tau \quad \forall j \in I_k\end{cases}$$

where $I_k = \{j \text{ s.t. } g_j(x_k) + \langle \nabla g_j(x_k), d_k\rangle = -|d_k^0|^\nu\}$.
If (LS) has no solution or if $|\tilde{d}_k| > |d_k|$, set $\tilde{d}_k = 0$.
Proceed to Step 2.

(iv) Compute a first order feasible descent direction d_k (see remark below).
Set $\theta_k = \langle \nabla f(x_k), d_k\rangle$.
Set $\tilde{d}_k = 0$.

Step 2. *Line search.*
Compute t_k, the first number t of the sequence $\{1, \beta, \beta^2, \cdots\}$ satisfying

$$f(x_k + td_k + t^2\tilde{d}_k) \leqq f(x_k) + \alpha t\theta_k, \tag{2.1}$$

$$g_j(x_k + td_k + t^2\tilde{d}_k) \leqq 0, \qquad j=1,\cdots m. \tag{2.2}$$

Step 3. *Updates.*
Compute a new approximation H_{k+1} of the Hessian matrix.
Set $x_{k+1} = x_k + t_k d_k + t_k^2 \tilde{d}_k$.
Set $k = k+1$.
Go back to Step 1. □

Remark. The "first order" direction of Step 1(iv) is any direction satisfying a set of conditions that will be stated later, as the need arises. At this time, let us just point out that algorithms do exist that construct directions satisfying these conditions (e.g. the algorithm in [16] using optimality function θ_ε^2 defined by equation (36) in that paper).

3. Global convergence. In this section we prove that, under mild conditions, the algorithm described in § 1 is convergent.

In addition to H1 and H2, we will assume that the following hypothesis holds.

H3. For any $x \in X$, the vectors $\{\nabla g_j(x), j \in I(x)\}$ are linearly independent, where

$$I(x) \triangleq \{j \text{ s.t. } g_j(x) = 0\}.$$

Before analyzing the convergence properties of Algorithm A, we need to verify that the line search of Step 2 is well defined. A first requirement on the first order direction is needed here.

R1. The direction computed at Step 1(iv) of the algorithm is a strict descent direction for f and for the active constraints associated with the current iterate (i.e., $\langle \nabla f(x_k), d_k \rangle < 0$ and $\langle \nabla g_j(x_k), d_k \rangle < 0$ for all $j \in I(x_k)$).

PROPOSITION 3.1. *The line search yields a step $t_k = \beta^j$ for some finite $j = j(k)$.*

Proof. This is a well-known result in the case when the direction is computed at Step 1(iv) and R1 is satisfied. Thus suppose that the direction is computed through Step 1(i)–(iii). We have

$$f(x_k + td_k + t^2\tilde{d}_k) = f(x_k) + t\langle \nabla f(x_k + \xi d_k + \xi^2 \tilde{d}_k), d_k + 2\xi \tilde{d}_k \rangle$$

for some $\xi \in [0, t]$. Since f is continuously differentiable, $\theta_k = \langle \nabla f(x_k), d_k \rangle < 0$ (from Step 1 (ii)), and $\alpha \in (0, \frac{1}{2})$, there exists $\underline{t} > 0$ such that

$$f(x_k + td_k + t^2\tilde{d}_k) \leqq f(x_k) + t\alpha\theta_k \quad \forall t \in [0, \underline{t}].$$

We also have

$$g_j(x_k + td_k + t^2\tilde{d}_k) = g_j(x_k) + t\langle \nabla g_j(x_k + \xi d_k + \xi^2\tilde{d}_k), d_k + 2\xi\tilde{d}_k \rangle$$

for some $\xi \in [0, t]$. Moreover, from the inequalities

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k \rangle \leqq -|d_k^0|^\nu < 0, \qquad j = 1, \cdots m$$

and

$$g_j(x_k) \leqq 0, \qquad j = 1, \cdots m,$$

we conclude that either $g_j(x_k) < 0$ or $g_j(x_k) = 0$ and $\langle \nabla g_j(x_k), d_k \rangle < 0$. Therefore, for $j = 1, \cdots m$, there exists some $\underline{t}_j$ such that

$$g_j(x_k + td_k + t^2\tilde{d}_k) \leqq 0 \quad \forall t \in [0, \underline{t}_j]. \qquad \square$$

It is of interest to note that this result was obtained without making use of any property of $\tilde{d}_k$.

Our first convergence result has to do with the sequence of intermediate directions $\{d_k^0\}$.

PROPOSITION 3.2. *Suppose that Algorithm* A *generates an infinite sequence. Let x^* be a cluster point of this sequence, and $\{x_k\}_{k \in K}$ a subsequence converging to x^*. Suppose moreover that the directions at points x_k, for $k \in K$, are computed through Step* 1(i)–(iii). *Then, the sequence $\{d_k^0\}_{k \in K}$ tends to zero.*

Proof. We assume by contradiction that there exists a cluster point x^*, a number $\underline{d} > 0$ and subsequences $\{x_k\}_{k \in K}$ and $\{d_k^0\}_{k \in K}$ such that

$$x_k \to x^*, \quad k \in K, \quad k \to \infty$$

and

$$|d_k^0| \geqq \underline{d} \quad \forall k \in K.$$

We first show that, in that case, the step t_k obtained by the line search is bounded away from zero on K, i.e.,

$$\exists \underline{t} > 0 \text{ s.t. } t_k \geqq \underline{t} \quad \forall k \in K. \tag{3.1}$$

From Step 1(ii) we have, for $k \in K$,

$$\theta_k = \langle \nabla f(x_k), d_k \rangle \leqq -(\underline{d})^\kappa$$

and

$$g_j(x_k)+\langle\nabla g_j(x_k),\, d_k\rangle \leqq -(\underline{d})^{\nu}.$$

Then, for $k\in K$, k large enough, we obtain,

$$\langle\nabla f(x_k),\, d_k\rangle \leqq -\delta$$

and

$$\langle\nabla g_j(x_k),\, d_k\rangle \leqq -\delta \quad \forall j\in I(x^*),$$
$$g_j(x_k)\leqq -\delta \quad \forall j\notin I(x^*)$$

for some $\delta>0$. From the identity

$$f(x_k+td_k+t^2\tilde{d}_k)=f(x_k)+\int_0^1 \langle\nabla f(x_k+t\xi d_k+t^2\xi^2\tilde{d}_k),\, td_k+2t^2\xi\tilde{d}_k\rangle d\xi,$$

it then follows that, for $k\in K$, k large enough,

$$\begin{aligned} f(x_k&+td_k+t^2\tilde{d}_k)-f(x_k)-\alpha t\theta_k \\ &\leqq t\left\{\int_0^1 [\langle\nabla f(x_k+t\xi d_k+t^2\xi^2\tilde{d}_k),\, d_k+2t\xi\tilde{d}_k\rangle-\langle\nabla f(x_k),\, d_k\rangle]\, d\xi\right. \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\left. +(1-\alpha)\langle\nabla f(x_k),\, d_k\rangle\right\} \\ &\leqq t\{\sup_{\xi\in[0,1]} |\nabla f(x_k+t\xi d_k+t^2\xi^2\tilde{d}_k)-\nabla f(x_k)|\,|d_k| \\ &\qquad\qquad +2t\sup_{\xi\in[0,1]}|\nabla f(x_k+t\xi d_k+t^2\xi^2\tilde{d}_k)|\,|\tilde{d}_k|-(1-\alpha)\delta\}. \end{aligned}$$

Since d_k and $\tilde{d}_k$ are bounded and $f\in C^1$, this ensures that there exists $\underline{t}_f>0$, independent of k, such that for $t\in[0,\underline{t}_f]$, $k\in K$, k large enough,

$$f(x_k+td_k+t^2\tilde{d}_k)-f(x_k)-\alpha t\theta_k\leqq 0.$$

Similarly, for $k\in K$, k large enough, $t>0$ and $j\in I(x^*)$, it holds

$$\begin{aligned} g_j(x_k+td_k+t^2\tilde{d}_k)-g_j(x_k)\leqq t\{&\sup_{\xi\in[0,1]}|\nabla g_j(x_k+t\xi d_k+t^2\xi^2\tilde{d}_k)-\nabla g_j(x_k)|\,|d_k| \\ &+2t\sup_{\xi\in[0,1]}|\nabla g_j(x_k+t\xi d_k+t^2\xi^2\tilde{d}_k)|\,|\tilde{d}_k|-\delta\} \end{aligned}$$

so that there exists some $\underline{t}_j>0$ independent of k such that, for $t\in[0,\underline{t}_j]$, $k\in K$, k large enough,

$$g_j(x_k+td_k+t^2\tilde{d}_k)\leqq 0.$$

Also, there exists $\underline{t}_j>0$ independent of k such that, for $t\in[0,\underline{t}_j]$, $k\in K$, k large enough, and $j\notin I(x^*)$,

$$g_j(x_k+td_k+t^2\tilde{d}_k)\leqq g_j(x_k)+\frac{\delta}{2}\leqq -\delta+\frac{\delta}{2}\leqq 0 \quad \forall t\in[0,\underline{t}_j].$$

Our claim (3.1) is thus proven, with $\underline{t}=\min\{\underline{t}_f,\underline{t}_j,\, j=1,\cdots m\}$.

Now, for $k\in K$, k large enough, we have,

$$\begin{aligned} f(x_{k+1})&\leqq f(x_k)+\alpha t_k\theta_k \\ &\leqq f(x_k)-\alpha\underline{t}\delta. \end{aligned} \tag{3.2}$$

On the other hand, from (2.1), the sequence $\{f(x_k)\}$ is monotonically decreasing and hence, since f is continuous, $f(x_k) \to f(x^*)$ as $k \to \infty$. This contradicts (3.2). □

In order to prove global convergence of Algorithm A, we need to strengthen the first requirement on the first order direction, replacing it with R1′.

R1′. If a subsequence $\{x_k\}_{k \in K}$ converges to a point x^* which is not a Kuhn–Tucker point for problem (P), then the corresponding first order directions are bounded and satisfy the inequalities

$$\langle \nabla f(x_k), d_k \rangle \leqq -\delta,$$

$$\langle \nabla g_j(x_k), d_k \rangle \leqq -\delta \quad \forall i \in I(x^*)$$

for all $k \in K$, for some $\delta > 0$.

THEOREM 3.3. *Algorithm* A *described in* § 2 *either stops at a Kuhn–Tucker point or generates a sequence* $\{x_k\}$ *for which each accumulation point is a Kuhn–Tucker point for* (P).

Proof. The first statement is obvious, the only stopping point being in Step 1(i). Thus, suppose that $\{x_k\}_{k \in K} \to x^*$. If the first order direction is selected infinitely many times, the result follows from an argument identical to that used in the proof of Proposition 3.2, using requirement R1′ and the fact that the function f is monotonically decreasing. We then suppose, without loss of generality, that the direction is always computed through Step 1(i)–(iii) on K and that the active set associated with (QP_0) keeps a constant value

$$I = I_k = \{j \text{ s.t. } g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle = 0\} \quad \forall k \in K.$$

From Proposition 3.2, we have

$$d_k^0 \to 0, \quad k \in K, \quad k \to \infty.$$

Therefore, $I \subset I(x^*)$. Also, the vector d_k^0 satisfies the optimality conditions

$$H_k d_k^0 + \nabla f(x_k) + \sum_{j \in I} (\mu_k)_j \nabla g_j(x_k) = 0, \tag{3.3a}$$

$$(\mu_k)_j \geqq 0, \tag{3.3b}$$

for some multiplier vector μ_k. Because $I \subset I(x^*)$, for $k \in K$, k large enough, the vectors $\nabla g_j(x_k)$, $j \in I$ are linearly independent. If we denote by $R_I(x_k)$ the $n \times |I|$ matrix

$$R_I(x_k) = (\nabla g_j(x_k) \text{ s.t. } j \in I)$$

we obtain the expression of the unique multiplier vector μ_k as

$$\mu_k = -(R_I^T(x_k) R_I(x_k))^{-1} R_I(x_k)^T (H_k d_k^0 + \nabla f(x_k)).$$

Due to the condition $|H_k d_k^0| \leqq |d_k^0|^{1/2}$ we obtain

$$\mu_k \to \mu^*, \quad k \in K, \quad k \to \infty$$

with

$$\mu^* = -(R_I^T(x^*) R_I(x^*))^{-1} R_I(x^*)^T \nabla f(x^*).$$

Taking the limit in (3.3) yields

$$\nabla f(x^*) + \sum_{j \in I} \mu_j^* \nabla g_j(x^*) = 0, \qquad \mu_j^* \geqq 0.$$ □

We conclude this section by showing that the existence of an accumulation point in the sequence generated by Algorithm A induces some regularity properties on this sequence. This result will be used in § 4. We first need to introduce a second and last requirement on the first order direction.[2]

R2. The first order direction satisfies the relation $\langle \nabla f(x_k), d_k \rangle \leqq -c|d_k|^\alpha$ for some $\alpha \geqq 1$ and $c > 0$.[3]

PROPOSITION 3.4. *Suppose that the sequence* $\{x_k\}$ *generated by Algorithm* A *has some accumulation point. Then*

$$\{|x_{k+1} - x_k|\} \to 0, \qquad k \to \infty.$$

Proof. Since $f(x_k)$ is monotonically decreasing, existence of an accumulation point of $\{x_k\}$ and continuity of f imply that the sequence $\{f(x_k)\}$ is bounded. Also, the line search in Algorithm A yields

$$f(x_{k+1}) \leqq f(x_k) + \alpha t_k \theta_k.$$

It follows that

$$t_k \theta_k \to 0, \qquad k \to \infty. \tag{3.4}$$

Now $t_k\theta_k$ is bounded from above by $-t_k|d_k|^\kappa$ if the direction is computed through Step 1(i)–(iii) and by $-ct_k|d_k|^\alpha$ if the direction is computed through Step 1(iv). Thus, in both cases, (3.4) and the fact that the step t_k is bounded by 1 imply

$$|t_k d_k| \to 0, \qquad k \to \infty.$$

Since

$$\begin{aligned} |x_{k+1} - x_k| &\leqq t_k|d_k| + t_k^2|\tilde{d}_k| \\ &\leqq 2t_k|d_k| \end{aligned}$$

the claim holds. □

4. Rate of convergence. In order to study the rate of convergence of the algorithm, we need some stronger regularity assumptions on the functions involved in problem (P). We replace H2 by the following hypothesis.

H2′. The functions f, g_j, $j = 1, \cdots m$ are three times continuously differentiable. Hypotheses H1 and H3 are still assumed to hold.

Let x^* be a Kuhn–Tucker point for (P). Denote by μ^* the unique multiplier vector computed at x^* and, for any $x \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^m$, denote by $L(x, \mu)$ the Lagrangian function

$$L(x, \mu) = f(x) + \sum_j (\mu)_j g_j(x).$$

The optimality conditions associated with x^* can then be written

$$\begin{gathered} \nabla_x L(x^*, \mu^*) = 0, \\ \mu^* \geqq 0, \qquad g_j(x^*) \leqq 0, \quad j = 1, \cdots m, \\ (\mu^*)_j g_j(x^*) = 0, \qquad j = 1, \cdots m. \end{gathered}$$

The point x^* is said to satisfy *second order sufficiency conditions with strict complementary slackness* if the multipliers satisfy $\mu_j^* > 0$ for all $j \in I(x^*)$ and if the Hessian of the Lagrangian function $\nabla_{xx} L(x^*, \mu^*)$ is positive definite on the subspace $\{p \text{ s.t. } \langle \nabla g_j(x^*), p \rangle = 0 \text{ for all } j \in I(x^*)\}$.

[2] We could replace R2 by any condition sufficient for Proposition 3.4 to hold.

[3] The direction computed by the method described in [16, eq. (36)] satisfies R2 with $\alpha = 2$ and $c = 1$.

PROPOSITION 4.1. *If some accumulation point x^* of the sequence generated by the algorithm satisfies the second order sufficiency conditions with strict complementary slackness, then the entire sequence converges to x^*.*

Proof. Under the stated assumptions, the Kuhn-Tucker point x^* is isolated (see, e.g., [19]), i.e., for some $\varepsilon > 0$, the ball $B(x^*, \varepsilon)$ does not contain any Kuhn-Tucker point other than x^*. From Proposition 3.4, we have

$$|x_{k+1} - x_k| \to 0, \qquad k \to \infty.$$

Therefore, for k large enough, $|x_{k+1} - x_k| < \varepsilon/4$ and there exists a subsequence $\{x_k\}_{k \in K}$ such that $|x_k - x^*| < \varepsilon/4$ on K. It is then impossible to leave $B(x^*, \varepsilon)$ without creating another cluster point and hence a Kuhn-Tucker point in that ball. □

In the sequel, we will assume that the sequence generated by the algorithm converges to such a point x^*. We will denote by R^* and P^* the $n \times |I(x^*)|$ and $n \times n$ matrices, respectively, defined by

$$R^* = \{\nabla g_j(x^*), j \in I(x^*)\},$$
$$P^* = I - R^*(R^{*T}R^*)^{-1}R^{*T}.$$

Given some iterate x_k close enough to x^*, we will similarly define matrices R_k and P_k by

$$R_k = \{\nabla g_j(x_k), j \in I(x^*)\},$$
$$P_k = I - R_k(R_k^T R_k)^{-1}R_k^T.$$

Without loss of generality, we will suppose that the matrices H_k are symmetric. We will assume moreover that the sequence $\{H_k\}$ converges to a matrix H^* satisfying

$$P^*H^*P^* = P^*\nabla_{xx}^2 L(x^*, \mu^*)P^*. \tag{4.1}$$

This holds, for example, when one uses secant approximations as in [9] or, under suitable conditions, when one uses the BFGS update formula (see [18]). Hypothesis (4.1) and the second order sufficiency condition guarantee the existence of a positive number ρ satisfying[4]

$$d^T P_k H_k P_k d \geqq \rho |P_k d|^2 \quad \forall d \in \mathbb{R}^n \tag{4.2}$$

for k large enough.

Propositions 4.2 and 4.3 give important asymptotic properties.

PROPOSITION 4.2. *For k large enough,*

(i) (QP_0) *has a unique Kuhn-Tucker point of least norm,*
(QP) *has a unique Kuhn-Tucker point of least norm,*
$\{d_k^0\} \to 0, \qquad \{d_k\} \to 0,$
where d_k^0 and d_k are computed through Step 1(i) *and* (ii).

(ii) $\{\mu_k^0\} \to \mu^*, \qquad \{\mu_k\} \to \mu^*$
where μ_k^0 and μ_k are the multipliers associated with the quadratic problems (QP_0) *and* (QP).

(iii) $I_k^0 \triangleq \{j \text{ s.t. } (\mu_k^0)_j > 0\} = \{j \text{ s.t. } g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle = 0\} = I(x^*),$

$I_k \triangleq \{j \text{ s.t. } (\mu_k)_j > 0\} = \{j \text{ s.t. } g_j(x_k) + \langle \nabla g_j(x_k), d_k \rangle = -|d_k^0|^\nu\} = I(x^*).$

[4] In fact if (4.2) holds, a positive matrix H_k' can easily be constructed such that $P_k H_k' P_k = P_k H_k P_k$ (see [18]).

Proof. x^* is a Kuhn–Tucker point for the problem

$$\min \tfrac{1}{2}(x-x^*)^T H^*(x-x^*) + \langle \nabla f(x^*), x-x^* \rangle$$
$$\text{s.t. } g_j(x^*) + \langle \nabla g_j(x^*), x-x^* \rangle \leqq 0$$

at which the second order sufficiency conditions are satisfied with strict complementary slackness and linear independence of the gradients of the active constraints.

We can write $d_k^0 = x - x_k$ where x is solution of the problem

$$\min \tfrac{1}{2}(x-x_k)^T H_k(x-x_k) + \langle \nabla f(x_k), x-x_k \rangle$$
$$\text{s.t. } g_j(x_k) + \langle \nabla g_j(x_k), x-x_k \rangle \leqq 0.$$

Since $x_k \to x^*$ and $H_k \to H^*$, parts (i) and (ii) for d_k^0 follow from Theorem 2.1 of [20]. We can also write $d_k = x - x_k$ where x is solution of the problem

$$\min \tfrac{1}{2}(x-x_k)^T H_k(x-x_k) + \langle \nabla f(x_k), x-x_k \rangle$$
$$\text{s.t. } g_j(x_k) + \langle \nabla g_j(x_k), x-x_k \rangle \leqq -|d_k^0|^\nu$$

and, as $d_k^0 \to 0$, parts (i) and (ii) for d_k also follow from Theorem 2.1 of [20]. That (iii) is true follows from the fact that $\mu_k^0 \to \mu^*$, $\mu_k \to \mu^*$, and that, from strict complementarity, $I(x^*) = \{j \text{ s.t. } \mu_j^* > 0\}$. □

PROPOSITION 4.3. *The solutions of* (QP$_0$) *and* (QP) *satisfy*

$$\{d_k^0\} \sim \{d_k\}, \tag{4.3}$$

i.e., there exist some constants $C_1 > 0$, $C_2 > 0$ *and an integer* $\hat{k}$ *such that*

$$C_1 |d_k^0| \leqq |d_k| \leqq C_2 |d_k^0| \quad \forall k \geqq \hat{k}.$$

Proof. For k large enough, $I_k^0 = I_k = I(x^*)$ and d_k satisfies

$$H_k d_k + \nabla f(x_k) + R_k \mu_k = 0. \tag{4.4}$$

Let us define Δd_k and $\Delta \mu_k$ by

$$d_k = d_k^0 + \Delta d_k, \qquad \mu_k = \mu_k^0 + \Delta \mu_k.$$

We have from (4.4)

$$H_k d_k^0 + \nabla f(x_k) + R_k \mu_k^0 + H_k \Delta d_k + R_k \Delta \mu_k = 0$$

which gives

$$H_k \Delta d_k + R_k \Delta \mu_k = 0. \tag{4.5}$$

Now, Δd_k solves

$$R_k^T \Delta d_k = \begin{pmatrix} -|d_k^0|^\nu \\ \vdots \\ -|d_k^0|^\nu \end{pmatrix}$$

and can be decomposed into

$$\Delta d_k = \Delta d_k^1 + \Delta d_k^2$$

with

$$\Delta d_k^1 = P_k \Delta d_k$$

and

$$\Delta d_k^2 = -R_k (R_k^T R_k)^{-1} \begin{pmatrix} |d_k^0|^\nu \\ \vdots \\ |d_k^0|^\nu \end{pmatrix}. \tag{4.6}$$

LEMMA 4.6. *There exists a positive constant $\tilde{\gamma}$ such that, for k large enough, the solution d_k of* (QP) *satisfies the inequality*

$$\theta_k = \langle \nabla f(x_k), d_k \rangle \leqq -\tilde{\gamma}|d_k|^2.$$

Proof. Direction d_k computed through Step 1(ii) satisfies the Kuhn–Tucker conditions

$$\nabla f(x_k) + \sum_{j \in I(x^*)} (\mu_k)_j \nabla g_j(x_k) + H_k d_k = 0$$

and, multiplying by d_k,

$$\theta_k = -\sum_{j \in I(x^*)} (\mu_k)_j \langle \nabla g_j(x_k), d_k \rangle - d_k^T H_k d_k$$

which yields, using the complementarity conditions,

$$\theta_k = \sum_{j \in I(x^*)} (\mu_k)_j g_j(x_k) + \sum_j (\mu_k)_j |d_k^0|^\nu - d_k^T H_k d_k.$$

Replacing d_k by its decomposition, we obtain

$$\begin{aligned}\theta_k = &\sum_{j \in I(x^*)} (\mu_k)_j g_j(x_k) + \sum_j (\mu_k)_j |d_k^0|^\nu - d_k^T P_k H_k P_k d_k \\ &- 2 d_k^T P_k H_k d_k^1 - d_k^{1^T} H_k d_k^1.\end{aligned}$$

Using (4.2), Lemmas 4.4 and 4.5, and the fact that the matrices H_k and the multipliers μ_k are bounded, we obtain

$$\theta_k \leqq -\rho |d_k|^2 + O(|d_k^0|^\nu).$$

Since $\{d_k\} \sim \{d_k^0\}$, the claim holds. □

The next proposition shows that, for k large enough, the algorithm never needs to compute a first order direction.

PROPOSITION 4.7. *For k large enough, the solutions of* (QP_0) *and* (QP) *satisfy the following inequalities*

(i) $|d_k^0| \leqq M,$

(ii) $|H_k d_k^0| \leqq |d_k^0|^{1/2},$

(iii) $|d_k| \leqq M,$

(iv) $\theta_k = \langle \nabla f(x_k), d_k \rangle \leqq \min\{-|d_k^0|^\kappa, -|d_k|^\kappa\}.$

Proof. Relations (i)–(iii) obviously hold since the sequences $\{d_k^0\}$ and $\{d_k\}$ converge to zero and the matrices H_k are bounded. Inequality (iv) follows from Lemma 4.6. □

A crucial requirement for achieving superlinear convergence is that a unit stepsize be used in a neighborhood of the solution. The next proposition shows that Algorithm A does achieve this goal.

PROPOSITION 4.8. *For k large enough, the direction is always computed through Step* 1(i)–(iii) *and the stepsize t_k is one.*

Proof. (The proof is analogous to that of Proposition 15 in [8]; see also [9].)

In all the relations given in this proof, the phrase "for k large enough" is implicit.

The first part of the theorem is obvious in view of Proposition 4.7. In order to prove the second part, we first show that the property

$$\tilde{d}_k = O(|d_k|^2)$$

holds (close to the solution, $\tilde{d}_k$ is always well defined). By definition, $\tilde{d}_k$ is the minimal norm solution of

$$g_j(x_k+d_k)+\langle\nabla g_j(x_k), \tilde{d}_k\rangle=-|d_k^0|^\tau, \qquad j\in I_k.$$

Expanding, we obtain

$$\begin{aligned}g_j(x_k)+\langle\nabla g_j(x_k), d_k\rangle+\langle\nabla g_j(x_k), \tilde{d}_k\rangle\\ +\tfrac{1}{2}\langle d_k, \nabla_{xx}g_j(x_k+\xi d_k)d_k\rangle=-|d_k^0|^\tau, \qquad j\in I_k\end{aligned}$$

for some $0\leqq\xi\leqq 1$. Hence, using the definition of I_k

$$R_k^T\tilde{d}_k=O(|d_k|^2).$$

Thus, $\tilde{d}_k$ solves the problem

$$\begin{aligned}&\min \tfrac{1}{2}|\tilde{d}_k|^2\\ &\text{s.t. } R_k^T\tilde{d}_k=O(|d_k|^2)\end{aligned}$$

and, since R_k is full column rank, is given by

$$\tilde{d}_k=R_k(R_k^TR_k)^{-1}O(|d_k|^2),$$

which proves our first claim. Now, according to Step 2 in Algorithm A, two conditions are needed for the line search to yield a unit stepsize, namely feasibility of the resulting point (2.2) and sufficient decrease (2.1). Expanding g_j around x_k+d_k we obtain, for $j\in I(x^*)$,

$$\begin{aligned}g_j(x_k+d_k+\tilde{d}_k)&=g_j(x_k+d_k)+\langle\nabla g_j(x_k+d_k), \tilde{d}_k\rangle+O(|d_k|^4)\\ &=g_j(x_k+d_k)+\langle\nabla g_j(x_k), \tilde{d}_k\rangle+O(|d_k|^3)\\ &=-|d_k^0|^\tau+O(|d_k^0|^3).\end{aligned} \tag{4.8}$$

The last term is negative since the sequence $\{d_k^0\}$ converges to zero. Thus the feasibility condition is satisfied.

We also have, since f is three times continuously differentiable,

$$f(x_k+d_k+\tilde{d}_k)=f(x_k)+\langle\nabla f(x_k), d_k\rangle+\langle\nabla f(x_k), \tilde{d}_k\rangle+\tfrac{1}{2}d_k^T\nabla_{xx}f(x_k)d_k+O(|d_k|^3).$$

The Kuhn-Tucker conditions

$$\nabla f(x_k)+H_kd_k+\sum_j(\mu_k)_j\nabla g_j(x_k)=0$$

and the complementarity relations imply

$$\tfrac{1}{2}\langle\nabla f(x_k), d_k\rangle=-\tfrac{1}{2}d_k^TH_kd_k-\sum_j(\mu_k)_j\langle\nabla g_j(x_k), d_k\rangle-\tfrac{1}{2}\sum_j(\mu_k)_jg_j(x_k)+O(|d_k^0|^\nu)$$

and

$$\langle\nabla f(x_k), \tilde{d}_k\rangle=O(|d_k|^3)-\sum_j(\mu_k)_j\langle\nabla g_j(x_k), \tilde{d}_k\rangle.$$

We obtain therefore,

$$\begin{aligned}f(x_k+d_k+\tilde{d}_k)-f(x_k)=\tfrac{1}{2}\theta_k&-\sum_j(\mu_k)_j\langle\nabla g_j(x_k), d_k\rangle\\ &-\sum_j(\mu_k)_j\langle\nabla g_j(x_k), \tilde{d}_k\rangle-\tfrac{1}{2}d_k^TH_kd_k+\tfrac{1}{2}d_k^T\nabla_{xx}f(x_k)d_k\\ &+O(|d_k|^\nu)+O(|d_k|^3)-\tfrac{1}{2}\sum_j(\mu_k)_jg_j(x_k).\end{aligned} \tag{4.9}$$

Now, since the g_j's are three times continuously differentiable, the relation

$$g_j(x_k + d_k + \tilde{d}_k) = O(|d_k|^\tau), \quad j \in I(x^*)$$

obtained in (4.8) yields, for $j \in I(x^*)$,

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k \rangle + \langle \nabla g_j(x_k), \tilde{d}_k \rangle + \tfrac{1}{2} d_k^T \nabla_{xx} g_j(x_k) d_k = O(|d_k|^\tau).$$

Hence,

$$\begin{aligned} -\sum_j (\mu_k)_j \langle \nabla g_j(x_k), d_k \rangle - \sum_j (\mu_k)_j \langle \nabla g_j(x_k), \tilde{d}_k \rangle \\ = \sum_j (\mu_k)_j g_j(x_k) + \tfrac{1}{2} \sum_j (\mu_k)_j d_k^T \nabla_{xx} g_j(x_k) d_k + O(|d_k|^\tau). \end{aligned}$$

Substituting those values into (4.9), we obtain

$$\begin{aligned} f(x_k + d_k + \tilde{d}_k) - f(x_k) = {} & \tfrac{1}{2} \theta_k + \tfrac{1}{2} \sum_j (\mu_k)_j g_j(x_k) \\ & + \tfrac{1}{2} d_k^T \left(\nabla_{xx} f(x_k) + \sum_j (\mu_k)_j \nabla_{xx} g_j(x_k) - H_k \right) d_k \\ & + O(|d_k|^\tau) + O(|d_k|^\nu). \end{aligned}$$

This, together with Lemmas 4.4 and 4.5, gives

$$\begin{aligned} & f(x_k + d_k + \tilde{d}_k) - f(x_k) - \alpha \theta_k \\ & \leqq (\tfrac{1}{2} - \alpha) \theta_k + \tfrac{1}{2} d_k^T P_k \left(\nabla_{xx} f(x_k) + \sum_j (\mu_k)_j \nabla_{xx} g_j(x_k) - H_k \right) P_k d_k \\ & \quad + O(|d_k|^\tau) + O(|d_k|^\nu). \end{aligned}$$

Due to the convergence of the projections of the approximate Hessian matrices, we obtain

$$f(x_k + d_k + \tilde{d}_k) - f(x_k) - \alpha \theta_k \leqq (\tfrac{1}{2} - \alpha) \theta_k + o(|d_k|^2).$$

In view of Lemma 4.6, the right-hand side of the last inequality is nonpositive. Thus the "sufficient decrease" condition is satisfied. □

THEOREM 4.9. *Under the stated assumptions, the convergence is two-step superlinear, i.e., the following relation holds*

$$\lim_{k \to \infty} \frac{|x_{k+2} - x^*|}{|x_k - x^*|} = 0.$$

Proof. The proof is similar to the one of [18, Thm. 1]. □

5. Implementation and computational results. Several implementation issues have to be addressed. First, the sequence $\{H_k\}$ of $n \times n$ matrices is thus far unspecified, subject only to requirement (4.1). While a secant approximation to the Hessian $\nabla^2_{xx} L(x_k, \mu_k)$ would be suitable, use of an update formula avoids many function evaluations. Under some assumptions, matrices H_k generated by the BFGS formula [18] are shown to satisfy (4.1). The latter option, with $H_0 = I$, was selected for our experiments. Second, the order in which the tests (2.1) and (2.2) are performed needs to be specified. In our implementation, in line with the premise that the objective function may not be defined outside the feasible set, (2.2) was tested first and (2.1) was tested only when (2.2) was satisfied. Third, Algorithm A as stated does not efficiently handle affine constraints. In our experiments, the correction corresponding to such constraints was set to zero in the right-hand sides of the constraints in (QP) and (LS).

TABLE 1
Computational results.

No	Code	NF	NDF	FV	VC	KT
12	VFO2AD	12	12	−.30000000E+02	.58E−09	.35E−07
	OPRQP	40	26	−.30000004E+02	.76E−05	.15E−09
	A	7	7	−.30000000E+02	.0	.12E−06
29	VF02AD	13	13	−.22627417E+02	.0	.16E−05
	OPRQP	64	39	−.22627421E+02	.56E−05	.10E−05
	A	14	10	−.22627417E+02	.0	.17E−06
30	VF02AD	14	14	.10000000E+01	.0	.56E−08
	OPRQP	18	18	.10000000E+01	.38E−08	.28E−09
	A	14	13	.10000000E+01	.0	.0
31	VF02AD	10	10	.60000000E+01	.27E−09	.12E−04
	OPRQP	24	22	.59999631E+01	.62E−05	.13E−06
	A	11	8	.60000000E+01	.0	.41E−06
33	VF02AD	5	5	−.40000000E+01	.0	.0
	OPRQP	43	39	−.40000000E+01	.32E−10	.0
	A	4	4	−.40000000E+01	.0	.0
34	VF02AD	8	8	−.83403245E+00	.15E−08	.0
	OPRQP	60	37	−.83403515E+00	.73E−05	.0
	A	9	8	−.83403245E+00	.0	.43E−08
43	VF02AD	12	12	−.44000000E+02	.35E−09	.75E−05
	OPRQP	31	24	−.44000013E+02	.79E−05	.19E−06
	A	9	9	−.44000000E+02	.0	.68E−04
57	VF02AD	4	4	.30646306E−01	.0	.0
	OPRQP	40	24	.28459078E−01	.89E−05	.89E−06
	A	33	19	.28459673E−01	.0	.20E−07
66	VF02AD	7	7	.51816327E+00	.39E−08	.57E−06
	OPRQP	18	17	.51815751E+00	.10E−04	.11E−10
	A	8	8	.51816324E+00	.0	.0
84	VF02AD	6	6	−.52803365E+07	.63E−01	.0
	OPRQP	43	5	−.55883016E+07	.68E+00	.22E+06
	A	4	4	−.52803389E+07	.0	.0
100	VF02AD	20	20	.68063006E+03	.76E−07	.29E−03
	OPRQP	49	31	.68063005E+03	.76E−05	.73E−08
	A	42	14	.68063006E+03	.0	.21E−03
113	VF02AD	15	15	.24306209E+02	.16E−0	.11E−03
	OPRQP	30	28	.24306193E+02	.13E−04	.11E−08
	A	18	14	.24306209E+02	.0	.17E−04
117	VF02AD	17	17	.32348679E+02	.36E−07	.28E−05
	OPRQP	41	40	.32348442E+02	.54E−05	.73E−06
	A	28	16	.32348679E+02	.0	.68E−04

No: number of the test problem in [6].
Code: name of the program.
NF: number of objective function evaluations.
NDF: number of gradient evaluations of the objective function.
FV: objective function value at the final point.
VC: sum of constraint violation, given by $\sum_{j=1}^{m} \max(0, g_j(x))$, at the final point.
KT: norm of Kuhn-Tucker vector (i.e. norm of the gradient of the Lagrangian function at the final point).

However, in order to avoid potential zigzagging, the right-hand side in the condition defining I_k in (LS) was not set to zero for the affine constraints, but rather the corresponding "=" sign was changed to a "≧". Finally, *scaling* can be introduced at various places in the algorithm, and values have to be selected for the various parameters. If the right-hand side in the constraints in (QP) is too big, d_k may not be a descent direction for f in the early iterations, while if it is too small, the stepsize may be truncated, due to infeasibility, until a very small neighborhood of the solution is reached. In our experiments, the right-hand side of the constraints in (QP) and of the condition defining I_k in (LS) was replaced by max $(-|d_k^0|^3, -10^{-2}|d_k^0|)$, which seems to often result in a satisfactory behavior on reasonably well scaled problems. For a similar reason, we replaced the right-hand side of the constraints in (LS) by max $(-|d_k^0|^{5/2}, -10^{-2}|d_k^0|)$. The right-hand side of the test on θ_k in Step 1(ii) was scaled by a small number. This test was always satisfied throughout our experiments. Finally, we used $\alpha = .3$, $\beta = .8$ and $M = \infty$.

Algorithm A was tested on fourteen of the seventeen problems in [6] which do not involve equality constraints but do include *nonlinear* inequality constraints, and for which a feasible initial point is provided. Problems numbered 67, 70 and 85 were discarded due to some disparity between function values we computed and those given in [6]. When tested on Problem 93, with the chosen values of the algorithm parameters, Algorithm A had to resort to the first order direction (Step 1(ii)) for the initial iterations due to infeasibility of (QP), thus making the performance of Algorithm A dependent on the choice of the first order method. Table 1 shows the results obtained on the thirteen remaining problems. The results obtained with Algorithm A are compared to the best results among those given in [6], i.e., those obtained with algorithms VF02AD and OPRQP. The format of this table is as in [6].

In most cases, Algorithm A is competitive with VF02AD. It always performs better than OPRQP. This is remarkable since neither VF02AD nor OPRQP enjoys properties (1.1) and (1.2). It could be argued that a comparison based only on the number of function evaluations unduly favors Algorithm A, which calls for the solution of up to two quadratic programs and one linear least squares problem at each iteration. However, as pointed out in the introduction, clever implementation should reduce the computational effort needed to solve these three problems to little more than that required for the solution of a single quadratic program. Also, in the context of engineering design problems, function evaluations typically require such extensive computation that time spent in solving quadratic programs can generally be regarded as negligible. Finally, the number of constraint function evaluations is not indicated in Table 1. Typically, the number of such evaluations will be somewhat larger for Algorithm A than for its contenders due to the Maratos effect avoidance scheme.

REFERENCES

[1] M. K. H. Fan, C. D. Walrath, C. Lee, A. L. Tits, W. T. Nye, M. Rimer, R. T. Grant and W. S. Levine, *Two case studies in optimization-based computer-aided design of control systems*, Proc. 24th IEEE Conf. on Decision and Control (December 1985), p. 1794.

[2] U. M. Garcia-Palomares and O. L. Mangasarian, *Superlinearly convergent quasi-Newton algorithms for nonlinearly constrained optimization problems*, Math. Programming, 11 (1976), pp. 1-13.

[3] C. Gonzaga, E. Polak and R. Trahan, *An improved algorithm for optimization problems with functional inequality constraints*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 49-54.

[4] S. P. Han, *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22 (1977), pp. 297-309.

[5] J. HERSKOVITS, *A two-stage feasible direction algorithm for nonlinear constrained optimization*, Math. Programming (1987), to appear.

[6] W. HOCK AND K. SCHITTKOWSKI, *Test examples for nonlinear programming codes*, in Lecture Notes in Econom. and Math. Systems, 187, Springer-Verlag, Berlin, New York., 1981.

[7] N. MARATOS, *Exact penalty function algorithms for finite dimensional and optimization problems*, Ph.D. thesis, Imperial College of Science and Technology, London, U.K. (1978).

[8] D. Q. MAYNE AND E. POLAK, *A superlinearly convergent algorithm for constrained optimization problems*, Computing and Control Publication 78/52, Imperial College of Science and Technology, London (1978).

[9] ———, *A superlinearly convergent algorithm for constrained optimization problems*, Math. Programming Study, 16 (1982), pp. 45-61.

[10] L. W. NAGEL, SPICE2: *A computer program to simulate semiconductor circuits*, Memo No. ERL-M520, Electronics Research Laboratory, University of California, Berkeley, CA, May 1975.

[11] W. T. NYE AND A. L. TITS, *An enhanced methodology for interactive optimal design*, Proc. 1983 IEEE International Symposium on Circuits and Systems (May 1983), pp. 1050-1051.

[12] W. T. NYE, DELIGHT: *An interactive system for optimization-based engineering design*, Ph.D. thesis, Department EECS, University of California, Berkeley, CA, 1983.

[13] W. T. NYE AND A. L. TITS, *An application-oriented, optimization-based methodology for interactive design of engineering systems*, Internat. J. Control, 43 (1986), pp. 1693-1721.

[14] E. POLAK, *Computational Methods in Optimization*, Academic Press, New York, 1971.

[15] E. POLAK AND D. Q. MAYNE, *An algorithm for optimization problems with functional inequality constraints*, IEEE Trans. Automat. Control, AC-21 (1976), pp. 184-193.

[16] E. POLAK, R. TRAHAN AND D. Q. MAYNE, *Combined phase* I-*phase* II *methods of feasible directions*, Math. Programming, 17 (1979), pp. 32-61.

[17] M. J. D. POWELL, *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis, Dundee, 1977, Lecture Notes in Math., 630, G. A. Watson, ed., Springer-Verlag, Berlin, New York, 1977, pp. 144-157.

[18] ———, *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in Nonlinear Programming 3, O. L. Mangasarian, R. R. Meyer and S. M. Robinson, eds., Academic Press, New York, 1978, pp. 27-63.

[19] S. M. ROBINSON, *A quadratically-convergent algorithm for general nonlinear programming problems*, Math. Programming, 3 (1972), pp. 145-156.

[20] ———, *Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms*, Math. Programming, 7 (1974), pp. 1-16.

[21] A. L. TITS, W. T. NYE AND A. SANGIOVANNI-VINCENTELLI, *Enhanced methods of feasible directions for engineering design problems*, J. Optim. Theory Appl., 51 (1986), pp. 475-504.

[22] R. B. WILSON, *A simplified algorithm for concave programming*, Ph.D. dissertation, Harvard University, Cambridge, MA, 1963.

[23] G. ZOUTENDIJK, *Methods of Feasible Directions*, Elsevier, Amsterdam, 1960.